1

Key Concepts

This chapter introduces you to the key concepts covered in this book. These topics provide a general understanding of the components that constitute theme building in Community Server. The topics that are covered in this chapter include:

- ❑ Themes in Community Server 2007
- ❑ Types of themes
- ❑ Rendering diagrams
- ❑ `SiteUrls.config` and URL rewriting
- ❑ Chameleon
- ❑ Dynamic configuration
- ❑ Content scrubbing

Themes in Community Server 2007

The themes in Community Server 2007 are different from those in previous versions of Community Server. Not only has the process of creating a theme been simplified, but also the number of files that require creation have decreased. There is now a new set of controls across the site that is consistent and straightforward to use. These are significant improvements, as less is required from the theme developer.

Theme Portability

One of the major advances in Community Server 2007 over previous versions is the portability of theme settings. Not only are you able to easily install and use a theme, but you are also able to easily import specific theme settings and use them. This allows anyone with Community

Server 2007 and later to be able to alter the presentation of a theme and then to share that alteration with the world. One of the important qualities of this mechanism is that you do not have to be familiar with .NET or theme development. Instead, a novice user is able to alter their site and share this alteration.

Aside from the portability of themes in Community Server 2007, themes are now much easier to create. There is a standard set of controls that can be added to theme pages that have a consistent set of properties. Furthermore, the theme controls in Community Server 2007 support implicit binding. This means that a theme developer does not need to know how to retrieve a blog in order to display content from a blog. Instead, he or she only needs to place a blog control on a page that would normally display blog details, and the control will take care of the reset. The power of the new controls is actually quite amazing.

Standard Control Features

The theme controls in Community Server 2007 are not limiting. Instead, you are able to extend them and create custom controls that are consistent with the current controls. The new controls have many useful features, some of them include the following (please note that this is not an exhaustive list and that there are actually many more standard features for the controls):

- **DisplayConditions** — You are able to toggle the visibility of most controls using conditions, which can include custom conditions that you create.
- **Actions** — You can implement actions that take place after a form or other control type successfully or unsuccessfully does something. For example, if you want to redirect the user to a new page after he or she logs in, you can do so by placing `GoToSiteUrlAction` in the SuccessAction template on the login form.
- **Tag** — You are able to surround most controls with an html tag whenever the control is rendered. For example, if you want to surround a `ResourceControl` with a `div` tag, you can simply set the value of the `Tag` property to `div`. There is an enumerator for the tag property that provides you with intellisense whenever you are setting it.

Theme File Reduction

Less is required from users in order to change the look of their site. It is now much easier to switch themes on a live site. This enables a site to alter its presentation in order to meet the needs of the community that it is enabling. Now in Community Server 2007 there are dynamic themes. These types of themes enable an administrator to easily change how a site looks all from the Control Panel. A site can also allow blog owners to alter the presentation of their blog from the Control Panel with great ease.

Types of Themes

Community Server 2007 ships with one less type of theme than previous versions. All of the applications on a default Community Server site, except for weblogs are integrated into a site theme. Previously the Photo Gallery was able to have individual themes created just for it. Now these theme files are part of the main site theme.

Aside from the site theme there is the option to create weblog themes. This allows blog owners to select the theme that they would like to use on their blog. As a result, a site can have users that have blogs that look different from one another.

The types of themes that are available in Community Server 2007 and later include the following:

- ❑ Site themes
- ❑ Blog themes

Both sets of themes are found in the `Themes` folder. This is located off the root directory of a Community Server installation. Inside of the themes directory is a specific folder for containing all of the blog themes named Blogs. Also inside of the root themes directory are folders for the different site themes. In Community Server 2007 there are two site themes, `default` and `leanandgreen`.

Later in this book you will learn how to create each of these types of themes. In Chapter 6, you will learn how to create a custom Blog theme. In Chapter 7 you will learn how to create a custom Site theme. Both of these chapters will assist you in understanding what is required to create a theme in Community Server.

SiteUrls and URL Rewriting

A central component to any Community Server site is URL rewriting and the `SiteUrls.config` file. Both of these combine to provide very powerful features for your site.

The `SiteUrls.config` file contains the entries for the pages on your site. This provides a central location for controlling where URLs point to on your site. For example, if you want to change the name of a page on your site or the folder that it resides in, you can do so by changing a single URL entry in this file. This prevents a site administrator from having to touch physical page files to update the location that a link points to.

The `SiteUrls.config` file also provides data used in URL rewriting. URL rewriting allows for a requested page to be physically located in a different path than where it appears to be. Whenever you request `http://yoursite.com/default.aspx`, you are actually accessing the page at `http://yoursite.com/themes/default/common/home.aspx`. As you can see, URL rewriting is essential for themes to work correctly.

The URL rewriting feature allows your site to have many different themes and allows it to switch between them and yet still retain the same URL structure publicly. If you change from the default theme to the `leanandgreen` site theme, then your page files are going to be located at a physically different location. However, the users accessing the site do not need to type in a different path to use the different theme.

Chameleon

Chameleon is the name of the new theme engine available in Community Server 2007 and later versions. It provides many of the topics that are discussed in this book. Prior to Chameleon, there were no `theme.config` files to provide dynamic configuration options. Also, there were no controls that provided the same level of ease for creating powerful themes.

One of the main differences between Community Server 2007 and previous versions in regard to themes is that the previous versions contained skin files, whereas the current version does not. As a result of this change there has been a dramatic decrease in the number of files required to create a Community Server theme.

Another huge advantage to using the new Chameleon controls is that they provide inherent data binding. This means that a theme developer does not have to know how to query and get posts in order to display them. Now whenever a `WeblogData` control is added to a page, it knows to only display the data for the weblog that is on the current page.

In Chapter 5, you will learn much more about Chameleon. Chapter 5 discusses the various controls that are provided with Chameleon. This is an essential chapter that will help you learn much more about Chameleon.

Request to Rendering Diagram

In order to help you understand how URL rewriting and theming works in Community Server it is useful to look at an individual request to a site. The important thing to realize is that Community Server uses a custom HTTP module that makes URL rewriting occur. This is necessary because the module allows for custom code to execute during the processing of the ASP.NET pipeline. Figure 1-1 presents the basic steps that occur to make URL rewriting and loading of the appropriate theme page.

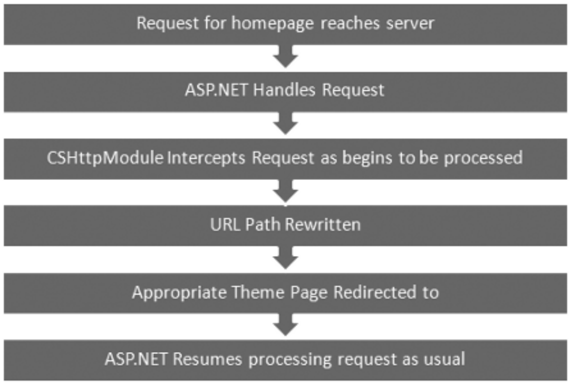
Request for homepage reaches server
ASP.NET Handles Request
CSHttpModule Intercepts Request as begins to be processed
URL Path Rewritten
Appropriate Theme Page Redirected to
ASP.NET Resumes processing request as usual

Figure 1-1: Page Request Including URL Rewriting Diagram

Whenever a request comes to a Community Server site that has the `CSHttpModule` enabled, the steps in Figure 1-1 take place. Early in the pipeline Community Server rewrites the requested path to the one configured by the URL data. More information about how the appropriate path is determined can be found in Chapters 3 and 4. Basically, though, at this point you should simply realize that specific URL requests are mapped to pages on a site.

Whenever this rewriting process occurs, the selected theme is included in the rewritten path. This means that if the `home.aspx` page is rewritten on a site using the default theme, the page at `Themes/default/Common/home.aspx` will be loaded. However, if the site were running the Lean and Green theme, the `Themes/leanandgreen/Common/home.aspx` page would be processed and returned to the client.

Therefore, in Figure 1-1 there are two separate steps for rewriting a page, even though they really do occur at the same time. The first is to determine the name of the page to actually load, and the other is to determine the theme that the page is found in.

Dynamic Configuration

One of the advances that were made with Community Server 2007 is the introduction of theme configuration files. These allow a theme's developer to provide various configuration options to control the presentation of a theme. As a result, end users of a dynamic theme are able to do more than use the theme; they are able to change the various ways that the theme is presented. For example, a configuration option could be changing the background color or image on a site. This allows users to be able to select these options from the Control Panel and control the presentation of their theme without having to physically alter any files.

A theme can allow for more advanced dynamic configuration options than background color or image choices. You can use the properties that can be created in a `theme.config` file to provide a way to toggle if controls are visible or not. Furthermore, you can allow for content to be entered in the theme configuration part of the Control Panel to be rendered in a specific location in your theme. There are numerous possibilities for what you can do with a dynamic theme.

Later in the book, in both Chapter 9 and Chapter 10, you will explore how to create these `theme.config` files. Additionally, you will learn many advanced concepts that are possible inside of these configuration files, such as creating custom theme rules. After reading these chapters, you should be more than comfortable with dynamic configuration options.

Content Scrubbing

In order to keep the presentation of the user-generated content well formed and alleviate security issues, Community Server uses several different content scrubbers. This means that the markup of posts on a site can be controlled. This is useful, as it can help maintain a consistent presentation as well as prevent unnecessary and potentially harmful markup from being delivered to users.

There are several different scrubbing modules that exist to perform scrubbing. Usually, though, only one is executed whenever a post is updated or created. This means that there are scrubbing modules that are specific to types of content. For example, the `WeblogPostandArticleHtmlScrubbing` module executes before a weblog post is committed to the database. This allows different HTML tags to exist on a weblog post, while not being allowed on a forum post.

Even though you can have different types of scrubbing occurring on a per application basis, in reality the scrubbing that occurs is mostly the same for each application. One of the nice features of the common `HtmlScrubber` is that it validates the HTML tags in a post. This is important, as it helps to allow your site be validated by the World Wide Web Consortium rules.

If you are interested in learning more about the current content scrubbing that takes place in Community Server, or even how to extend this feature, you can read Chapter 11. This chapter also teaches you about well-formed markup and how to maintain this across your site.

Summary

In this chapter, you were introduced to several different concepts. These concepts included the theming changes that exist in Community Server 2007, URL rewriting, and Content Scrubbing. In later chapters you will learn much more about each of these topics, as well as many more. If you have any questions or are interested in one of these topics and want to learn more about it you can go directly to the chapter or chapters discussing the topic.

In the next chapter, you will learn about how to get started developing for Community Server. You will learn how to set up your development environment and begin using some of the development tools to assist you in creating a theme.